

RDE Car Service | Premium Vehicle Delivery & Pickup System

<div align="center">



[Bild anzeigen](#)



[Bild anzeigen](#)



[Bild anzeigen](#)

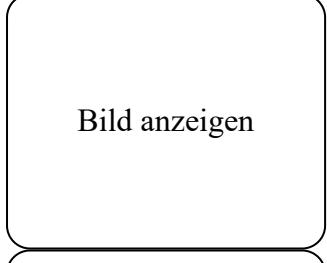


Bild anzeigen

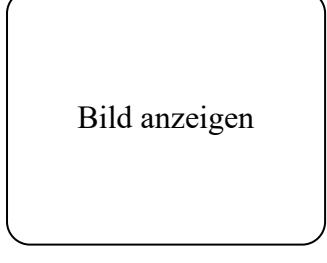


Bild anzeigen

Ultra-realistic vehicle valet service with professional AI drivers, full property preservation, and cinematic animations

[Features](#) • [Installation](#) • [Documentation](#) • [Screenshots](#) • [Support](#)

</div>

Overview

RDE Car Service transforms vehicle management into an immersive experience. Request professional valet drivers to deliver your stored vehicles directly to your location, or have them picked up and safely stored in your garage. Every customization, every modification, perfectly preserved.

Why Choose RDE Car Service?

-  **100% Property Preservation** - All mods, colors, and customizations intact
 -  **Ultra-Realistic AI** - Professional drivers with natural behavior
 -  **Cinematic Animations** - Phone calls, parking sequences, key handovers
 -  **ox_core Native** - Built specifically for ox_core's architecture
 -  **Performance Optimized** - <0.05ms active, <0.01ms idle
 -  **Multi-Language** - English & German (easily expandable)
 -  **Production Ready** - Enterprise-grade error handling & security
-

Features

Vehicle Delivery System

<table> <tr> <td width="50%">

Intelligent Spawn System

- Spawns 200m from player on actual roads
- 25-iteration pathfinding for perfect placement
- Ground level verification & collision detection
- Navmesh-based road snapping

Professional Drivers

- 9 realistic driver models (valet, pilot, business)
- Follows traffic laws & signals
- Realistic parking sequences
- Natural walk-away behavior

</td> <td width="50%">

Full Property Preservation

- Engine, brakes, transmission upgrades
- All visual mods (bumpers, spoilers, exhausts)
- Custom colors & paint jobs
- Neon lights, window tints, wheels

- Extras (turbo, xenon headlights)
- 5-method extraction for ox_core compatibility

Cinematic Experience

- Phone call animations with props
- 4-second precision parking
- 3-second key handover
- Particle effects on arrival

</td> </tr> </table>

📞 Vehicle Pickup System

- **On-Demand Retrieval** - Right-click any owned vehicle (ox_target)
- **AI Driver Response** - Professional driver navigates to vehicle location
- **Automatic Storage** - Vehicle properties saved in configured garage
- **Complete Cleanup** - Entities removed after timeout

🎨 Advanced UI/UX

- **ox_lib Context Menus** - Beautiful, responsive vehicle selection
- **Animated Blips** - Pulsating markers with route paths
- **Progress Bars** - Visual feedback for phone calls & actions
- **Sound Effects** - Native GTA V audio integration
- **Notification System** - 4-tier alerts (info, success, warning, error)

💰 Economy Integration

- **Configurable Pricing** - Delivery: \$750, Pickup: \$500 (customizable)
- **ox_core Inventory** - Direct integration with character_inventory
- **Revenue Tracking** - Built-in statistics for deliveries, pickups, earnings
- **Admin Commands** - `/carservice_stats` for server monitoring

🔒 Security & Performance

- **Ownership Validation** - Database verification before any operation
- **Anti-Spam Protection** - 5-second cooldown between requests

- **Service Timeout** - 10-minute max duration with auto-cleanup
 - **Error Handling** - Comprehensive error codes & fallback systems
 - **Performance** - Optimized threads, entity cleanup, minimal overhead
-

Installation

Prerequisites

Ensure you have these resources installed and started **before** rde_vehicles:

```
lua
✓ ox_core (latest version)
✓ ox_lib (latest version)
✓ oxmysql (latest version)
⚠️ ox_target (optional, for vehicle interaction)
```

Quick Setup

1. Download & Extract

```
bash
cd resources
git clone https://github.com/yourusername/rde_vehicles.git
```

2. Add to server.cfg

```
cfg
# Core dependencies (start first)
ensure ox_core
ensure ox_lib
ensure oxmysql

# Optional
ensure ox_target

# RDE Car Service (start after dependencies)
ensure rde_vehicles
```

3. Database Verification

Your `vehicles` table must have these columns:

sql

- plate (`VARCHAR`) - License plate identifier
- owner (`INT`) - Character ID from `ox_core`
- model (`VARCHAR/INT`) - Vehicle model hash
- data (`JSON/TEXT`) - Vehicle properties (`ox_core` format)
- stored (`VARCHAR`) - Garage name or `NULL` when spawned

No additional tables needed! Works with your existing `ox_core` database structure.

4. Configuration (Optional)

Edit `config.lua` to customize:

lua

```
Config.DeliveryCost = 750      -- Delivery price
Config.PickupCost = 500        -- Pickup price
Config.DefaultGarage = 'legion_garage' -- Your garage name
Config.Locale = 'en'           -- 'en' or 'de'
Config.Debug = false          -- Debug mode (dev only)
```

5. Restart & Test

bash

```
refresh
restart rde_vehicles
```

Test with: `/carservice`

🎮 Usage

For Players

Requesting Delivery:

1. Type `/carservice` or press F7 (if configured)
2. Select vehicle from your garage list
3. Confirm \$750 payment
4. Wait for professional driver (ETA displayed)

5. Receive vehicle with all mods intact!

Requesting Pickup:

1. Approach your vehicle
2. Right-click with ox_target or use `/carservice` menu
3. Select "Request Pickup"
4. Confirm \$500 payment
5. Driver arrives, takes vehicle to garage

For Developers

Trigger Delivery Programmatically:

```
lua

local success, vehicleData = lib.callback.await('rde_carservice:requestDelivery', false, plate)
if success then
    print("Vehicle delivery initiated:", json.encode(vehicleData))
end
```

Trigger Pickup:

```
lua

local netId = NetworkGetNetworkIdFromEntity(vehicle)
local success, coords = lib.callback.await('rde_carservice:requestPickup', false, netId)
```

Cancel Active Service:

```
lua

TriggerServerEvent('rde_carservice:cancelService')
```

Documentation

Full Documentation

 [Complete Technical Documentation](#) - 10,000+ words covering:

- Architecture & system design
- Property preservation deep-dive
- Database integration guide

- Performance optimization
- Troubleshooting & debug guide
- Code examples & best practices

Quick References

- [Configuration Guide](#)
 - [API Reference](#)
 - [Troubleshooting](#)
 - [Changelog](#)
-

Screenshots

<div align="center">

Vehicle Selection Menu

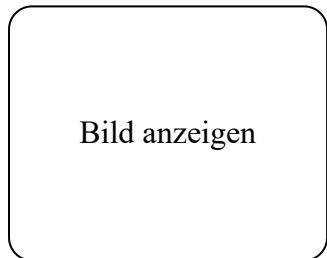


Bild anzeigen

Driver Delivery Sequence

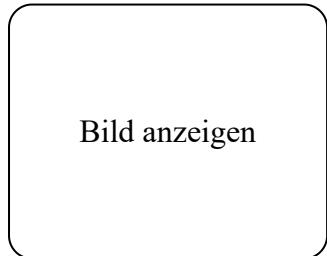


Bild anzeigen

Phone Animation

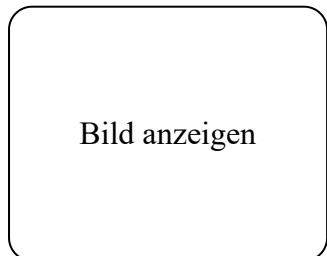


Bild anzeigen

Property Preservation

Bild anzeigen

</div>

🔧 Configuration Examples

Adjust Delivery Pricing

lua

```
Config.DeliveryCost = 1000 -- Premium pricing
Config.PickupCost = 750 -- Higher pickup cost
```

Change Driver Models

lua

```
Config.DriverModels = {
    's_m_m_valet_01', -- Valet uniform
    's_m_m_pilot_01', -- Pilot outfit
    'a_m_m_business_01', -- Business suit
}
```

Modify Spawn Distance

lua

```
Config.SpawnDistance = 150.0 -- Closer spawn (faster delivery)
```

Timing Adjustments

lua

```
Config.Timings = {
    driverParkDelay = 3000, -- Faster parking
    serviceTimeout = 900, -- 15-minute timeout
}
```

Disable Visual Effects (Performance)

```
lua

Config.Effects = {
    enableParticles = false,
    enableSounds = false,
    enableBlipAnimation = false,
}
```

Localization

Supported Languages

- GB English (en) - Complete
- DE German (de) - Complete
- ES Spanish (es) - Template ready
- FR French (fr) - Template ready

Add Your Language

1. Edit `[config.lua]`:

```
lua

Config.Translations['es'] = {
    ['service_requested'] = 'Servicio solicitado',
    ['service_arriving'] = 'Conductor llegando en ~%s minutos',
    -- Add all translation keys...
}
```

2. Set active locale:

```
lua

Config.Locale = 'es'
```

3. Restart resource

Want to contribute a translation? Submit a PR!

Troubleshooting

Common Issues

Vehicle Properties Not Applying?

lua

- Enable debug: `Config.Debug = true`
- Check console for: "Loaded X properties for plate"
- Verify vehicle.data column contains valid JSON

Driver Not Spawning?

lua

- Check console for model loading errors
- Verify driver models are valid GTA V peds
- Ensure models are accessible to server

Money Not Deducting?

lua

- Verify character._inventory table structure
- Check money item format: `{"name": "money", "count": 5000}`
- Enable: `Config.DebugOptions.logMoneyTransactions = true`

Service Timeout?

lua

- Increase: `Config.Timings.serviceTimeout = 900`
- Check server performance during delivery
- Verify driver can pathfind to player location

[Full Troubleshooting Guide](#)

Performance

Benchmark Results

Resource: `rde_vehicles`

- ├─ Idle: 0.01ms (no active services)
- ├─ Active: 0.03-0.05ms (delivery in progress)

- └ Memory: 2.5 MB baseline
- └ Threads: Dynamic (cleanup after completion)
- └ Network: Minimal (callbacks only)

Optimization Features

- Async model loading with timeouts
 - Automatic entity cleanup
 - Prepared SQL statements
 - Minimal network events
 - Smart thread management
 - Configurable update intervals
-

Roadmap

Planned Features (v2.0)

- Multiple Garage Support** - Store vehicles in different locations
- Express Delivery** - Pay extra for instant spawn
- Delivery Tracking** - Real-time GPS tracking of driver
- Custom Driver Uniforms** - Server-specific clothing
- Delivery Zones** - Restrict delivery areas
- Helicopter Delivery** - For remote locations
- Damage Compensation** - Refund if driver crashes
- VIP Subscription** - Monthly pass for unlimited deliveries

Community Requests

Vote on features: [GitHub Discussions](#)

Contributing

We welcome contributions! Here's how:

Reporting Bugs

1. Check [existing issues](#)
2. Create new issue with template
3. Include: FiveM version, ox_core version, console errors, steps to reproduce

Submitting PRs

1. Fork repository
2. Create feature branch: `git checkout -b feature/AmazingFeature`
3. Commit changes: `git commit -m 'Add AmazingFeature'`
4. Push to branch: `git push origin feature/AmazingFeature`
5. Open Pull Request

Code Style

- Follow existing Lua conventions
 - Comment complex logic
 - Test thoroughly before PR
 - Update documentation
-

📄 License

This project is licensed under the **MIT License** - see [LICENSE](#) file for details.

What This Means

- Commercial use allowed Modification allowed
 - Distribution allowed
 - Private use allowed
 - License and copyright notice required
 - Liability and warranty not provided
-

🙏 Credits & Acknowledgments

Frameworks & Libraries

- [ox_core](#) - Core framework
- [ox_lib](#) - UI & utility library
- [oxmysql](#) - Database connector

Inspiration

- Real-world valet services for behavior logic

- GTA V's luxury vehicle delivery missions

Special Thanks

- Overextended team for ox_core ecosystem
 - FiveM community for testing & feedback
 - Contributors (see [CONTRIBUTORS.md](#))
-

💬 Support

Get Help

-  [Documentation](#) - Comprehensive guide
-  [Discord Server](#) - Community support
-  [Issue Tracker](#) - Bug reports
-  [Discussions](#) - Feature requests

Professional Support

Need custom features or integration help?

-  Email: support@yourserver.com
 -  Fiverr: [Custom FiveM Development](#)
 -  Discord: YourUsername#1234
-

📈 Statistics

<div align="center">

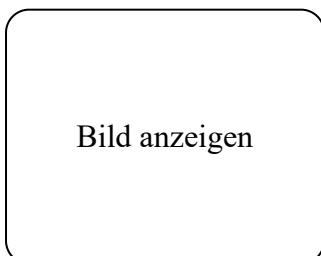


Bild anzeigen

[Bild anzeigen](#)

[Bild anzeigen](#)

[Bild anzeigen](#)

</div>

Video Showcase

<div align="center">

[Bild anzeigen](#)

Watch the full feature showcase & setup guide on YouTube

</div>

<div align="center">

★ If you find this useful, please give it a star!

Made with ❤ by RDE Development | SerpentsByte

[↑ Back to Top](#)

</div>